



**QUEEN'S
UNIVERSITY
BELFAST**

Digital ecclesia: Towards an online direct-democracy framework

Athanasopoulos, D. (2018). Digital ecclesia: Towards an online direct-democracy framework. In *Proceedings - 2018 ACM/IEEE 40th International Conference on Software Engineering: Software Engineering in Society, ICSE-SEIS 2018* ACM. <https://doi.org/10.1145/3183428.3183432>

Published in:

Proceedings - 2018 ACM/IEEE 40th International Conference on Software Engineering: Software Engineering in Society, ICSE-SEIS 2018

Document Version:

Peer reviewed version

Queen's University Belfast - Research Portal:

[Link to publication record in Queen's University Belfast Research Portal](#)

Publisher rights

Copyright 2018 ACM. This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

General rights

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Digital Ecclesia: Towards an Online Direct-Democracy Framework

ABSTRACT

Citizens currently envision the transition to the online direct democracy. However, the existing frameworks support the participatory democracy. Inspired by the Athenians' direct-democracy, we propose the initial version of the framework Digital-Ecclesia that offers dynamic citizens' engagement for achieving their universal participation. Furthermore, the Digital Ecclesia makes fair decisions in the case of conflicting votes by applying a game-theoretic method. For achieving privacy and scalability in the decision-making, the architecture of the Digital Ecclesia is distributed, i.e. each node is modelled as an autonomous software-component. To evaluate the fairness of the decision-making in the Digital Ecclesia, we conduct experiments on a corpus of real-world citizens' choices.

KEYWORDS

Distributed framework, message-passing model, conflicting choices, game theory.

ACM Reference format:

. 2018. Digital Ecclesia: Towards an Online Direct-Democracy Framework. In *Proceedings of International Conference on Software Engineering, Gothenburg, Sweden, May 2018 (ICSE-SEIS'18)*, 4 pages. DOI: 10.475/123.4

1 INTRODUCTION

Citizens increasingly seek electronic ways to cooperate for resolving their emerging issues, envisioning the transition from the traditional representative democracy to the online direct democracy. Citizens' attempts have recently led to the adoption of software frameworks for online democracy by public organizations (e.g. open ministry in Finland, participatory budgeting in Paris) [1].

Limitations & challenges. A framework supports the direct democracy if it enables the universal and direct citizens' participation [2], as was happening in Ecclesia (the sovereign governing body) of the Ancient Athenians' democracy¹ [3]. However, the existing frameworks support the participatory democracy, since they are used by citizens, who have previously asked their participation in specific working-groups. These frameworks do not ensure that relevant citizens are reached and prompted to engage (*on-demand engagement*). Towards supporting the direct democracy, one of the core challenges is the *dynamic engagement* of citizens.

However, the impact of the universal participation is that as the citizens' number increases, the diversity of their choices usually grow,

leading to *conflicts*. Ecclesia was guaranteeing the fair decision-making by compromising conflicting choices. On the contrary, the current frameworks make decisions assuming that *single consensus* (i.e. a single decision that represents the majority of citizens) can always be reached, violating the impossibility theorem for social decisions [4]. According to that theorem, single consensus on the common will cannot be manufactured out of citizens' choices. In that case, we face the challenge of compromising conflicting choices making *fair decisions*, i.e. they represent the majority of citizens.

Contribution. To address these challenges, we propose the framework *Digital Ecclesia*, inspired by the Ancient Athenians' democracy. The initial version of our framework offers dynamic citizens' engagement towards achieving universal and direct participation. To do so, the Digital Ecclesia is modelled as a social network that offers *high and dynamic reachability* of citizens. In particular, citizens are dynamically engaged via receiving notifications for new working groups. Notifications are broadcasted in the network with *encrypted messages*, starting from nodes that offer high reachability of the network [5]. Since each network node notifies only its neighbours, our broadcast is time and message efficient.

In the case of high participation, the Digital Ecclesia makes decisions from a large number of citizens' votes. For resolving privacy and scalability issues in the decision-making, the architecture of the Digital Ecclesia is *distributed* [6]. In particular, each node is an autonomous software-component that has *local storage* (e.g. votes are stored only locally) and *executes in parallel* the segments of the decision-making procedure that are related to locally-stored information. Thus, the Digital Ecclesia goes beyond the existing social-networking frameworks, whose nodes are simple user-profiles.

To make fair decisions, the Digital Ecclesia compromises conflicts by employing game-theoretic methods [7], which have not been adopted by the existing frameworks. According to [8], a decision compromises conflicting choices if it maximizes the Nash *social-welfare* function, assuming that citizens vote independently to each other and their choices are proportional to their preferences. To realize independent voting, we model the voting procedure as a *non-cooperative game* [9]. We further specify a distributed algorithm for employing the voting game. To ensure that citizens' votes are proportional to their preferences, we finally propose a metric that assesses the semantic similarity of past voting topics.

Concluding, the current version of the Digital Ecclesia offers a distributed algorithm for supporting the two (communication & engagement and decision-making) out of the five key-functions need to be addressed in democracy frameworks [2]. To evaluate the fairness of the decision-making in the Digital Ecclesia, we conduct experiments on a corpus of real-world citizens' choices.

The rest of the paper is structured as follows. Section 2 formally defines the notion of the voting game. Section 3 specifies our distributed framework. Section 4 presents the preliminary evaluation of our algorithm. Section 5 describes related approaches. Section 6 summarizes our contribution and discusses its future directions.

¹ It was the first time in the human history that the direct democracy was applied.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ICSE-SEIS'18, Gothenburg, Sweden

© 2018 Copyright held by the owner/author(s). 123-4567-24-567/08/06...\$15.00
DOI: 10.475/123.4

2 VOTING GAME

A game generally consists of a set of players (e.g. voters). Players make choices with a pay-off that depends on their utility functions [7]. We assume in the Digital Ecclesia voters provide a numerical choice that belongs to a scale of discrete values (e.g. 5-star scale). We further assume that the utility function of a voter is proportional to his/her (probabilistic) voting preferences. The latter is calculated based on voters' past choices on similar voting topics. Overall, we define below the notion of a voting game G for a topic p as follows.

Definition 2.1 (Voting game). A voting game is defined by the tuple $G = (T, N, R, U, W)$, such that:

- $T = \{t_1, t_2, \dots\}$ is a finite set of textual terms for a topic p ;
- $N = \{1, 2, \dots\}$ is a finite set of voters²;
- $R = \{1, 2, \dots\}$ is a finite set of numerical choices;
- $U_n(r) = k * pr_n(r)$ is the utility function of a voter n for a choice r and is proportional to his/her preferences pr_n with respect to the choice r (k is the constant of proportionality);
- $W(r) = \sum_{i=1}^N \log U_i(r)$ is the value of the social-welfare function for a choice r over the whole set of voters. \square

Definition 2.2 (Voting preference). The preference pr_n of a voter n on a choice r equals to the percentage of the times that s/he has made the choice r for a set s of similar topics over the total number of his/her all choices for these topics:

1. $\forall r \in R, pr_n(r) = \frac{v_n(r, s)}{v_n(s)} \in [0, 1]$,
 - a. $v_n(r, s)$ is the number of times that a voter n has made a choice r for a set s of similar topics;
 - b. $v_n(s)$ is the total number of the choices that s/he has made for a set s of similar topics;
 - c. $s = \{p_1, p_2, \dots, p_{|s|}\} : \forall p_i \in s, sim(p, p_i) > \theta \mid \theta \in [0, 1]$, is the set of the past topics p_i whose similarities to the current topic p are greater than a threshold θ ;
 - d. $sim(p_i, p_j) = \frac{\sum_{k=1}^{|T_1|} sim_t(t_k, f(t_k))}{|T_1|}, t_k \in T_1 \wedge f(t_k) \in T_2 \wedge |T_1| \leq |T_2|$ is the semantic similarity among the terms T_1 and T_2 of two topics p_i and p_j :
 - i. $f : t_k \rightarrow t_m \mid \sum_{k=1}^{|T_1|} sim_t(t_k, f(t_k))$ is max;
 - ii. $sim_t(t_k, t_m) = sim_{Lin}(t_k, t_m)$.
2. $\sum_{i=1}^{R_{max}} pr_n(r_i) = 1$, the sum of the rating probabilities of a consumer in the rating space R equals to 1. \square

Concerning the metric $sim(p_i, p_j)$, it calculates the similarity between two topics as the average of the similarities of their most similar pairs of terms. To do so, the metric solves the assignment problem f of the maximum-weighted matching in a bipartite graph [10], where the nodes correspond to terms and the edges to their similarities. Regarding the metric $sim_t(t_k, t_m)$, it assesses the relatedness of terms by calculating the Lin 's similarity of the corresponding ontology concepts in WordNet [11].

3 DISTRIBUTED FRAMEWORK

As previously discussed, the Digital Ecclesia is modelled as a bidirectional network of software components that communicate to each other by passing encrypted messages. Each component corresponds

to a single citizen, who holds authentication credentials. The *topology* of the network (a.k.a. communication links) can be dynamically updated during its lifetime³. We call neighbours the nodes connected with a direct communication-link.

To offer dynamic engagement and distributed voting game, the Digital Ecclesia includes the distributed algorithm in Section 3.1, whose time and message complexity are discussed in Section 3.2.

3.1 Distributed Algorithm

Our algorithm (Alg. 1) is triggered by a node (hereafter called source) that corresponds to one of the citizens who initiated a new working group. The algorithm continues to the voting procedure only if the source offers high reachability of the network. To cope with the dynamicity of the network (e.g. communication links may have been updated), the reachability of the source is dynamically calculated, as explained in the following.

The algorithm consists of a *local program* for each node that provides the ability to nodes to perform local computations and send/receive messages. Since nodes do not usually have the same computation clock, messages are sent *asynchronously*⁴. Nodes may send/receive the messages, $\langle reach \rangle$, $\langle parent \rangle$, $\langle source \rangle$, $\langle vote \rangle$, and $\langle utility \rangle$ (Alg. 1 (1, 9, 11, 17, and 32)).

Reach message. A reach message is initially sent by the source node to all its neighbours (Alg. 1 (2-4)). Every node that receives the message sets its parent node, sends parent and source messages (see below), and forwards the reach message to its neighbours except its parent (Alg. 1 (6-8)). Thus, a reach message is gradually forwarded (neighbour-by-neighbour) to all the nodes that are reachable from the source node (*broadcast* in message-passing systems [12]). To broadcast a reach message in a controlled way (i.e. messages do not wander in the network via different paths), a node does not forward messages that have already received (Alg. 1 (5)).

Source and parent messages. Since every node maintains its parent and children (distributed fashion), the algorithm constructs a *minimum spanning-tree* [13] (rooted at the source node) of the network part that is reachable from the source node. Moreover, every source-message is gradually forwarded back to the source node (Alg. 1 (16)) – *convergecast* in message-passing systems [12]. By receipt of source message, the source node increases the number of the received messages (Alg. 1 (13)). If the number is greater than a threshold ω , then the source node has high reachability⁵ and sends vote messages to its children (Alg. 1 (14-15)). Note that the source node does not send vote messages until the time limit has been expired in order to avoid to send them multiple times.

Vote message. When a node receives a vote message, then the node forwards the message to its children (Alg. 1 (18)). Following, the node asks by the citizen to vote for the current topic (Alg. 1 (21)), provided his/her credentials (Alg. 1 (19)). If the authentication is not successful, then the execution of the node for this topic is stopped (Alg. 1 (20)). Otherwise, the current vote of the citizen is locally stored, meeting our *privacy* requirement (Alg. 1 (29)). Following, the node updates the voting preferences of the citizen (Alg. 1 (23-28)) – applying Def. 2.2 – and sends the set of its

³We leave as future work the topology management of the Digital Ecclesia.

⁴For practical reasons, we considered in the implementation of the Digital Ecclesia upper bounds on voting time and message delays.

⁵In the implementation of the Digital Ecclesia, we used the threshold value 0.6.

²The framework assigns to each voter a unique identifier after his/her registration.

Algorithm 1 Dynamic Engagement & Distributed Voting Game// Code for node n_i , $1 \leq i \leq M$ (the total number of network nodes).**Input:** n_r, θ, ω

```

1: upon receiving  $\langle reach, p, R \rangle$  from neighbour  $n_j$  do
2:   if  $n_i = n_r$  then                                // It corresponds to the source node.
3:      $parent \leftarrow n_i$ 
4:     SEND  $\langle reach, p, R \rangle$  to all neighbours
5:   else if  $parent = \text{null}$  then
6:      $parent \leftarrow n_j$ 
7:     SEND  $\langle parent \rangle$  and  $\langle source \rangle$  to  $parent$ 
8:     SEND  $\langle reach, p, R \rangle$  to all neighbours except  $n_j$ 

9: upon receiving  $\langle parent \rangle$  from neighbour  $n_j$  do
10:   $children \leftarrow children \cup \{n_j\}$ 

11: upon receiving  $\langle source \rangle$  do
12:  if  $n_i = n_r$  then
13:     $++reached$ 
14:    if  $reached > \omega$  and  $expired = \text{true}$  then
15:      SEND  $\langle vote(p, R, \theta) \rangle$  to  $children$ 
16:    else SEND  $\langle source \rangle$  to  $parent$ 

17: upon receiving  $\langle vote(p, R, \theta) \rangle$  do
18:  SEND  $\langle vote(p, R, \theta) \rangle$  to  $children$ 
19:   $correct \leftarrow \text{AUTHENTICATE}$ 
20:  if  $correct = \text{false}$  then RETURN end if
21:   $r \leftarrow \text{VOTE}(p, R, \theta)$ 
22:   $\text{STORE}(p, r)$ 
23:  for all  $p_i \in (R_n - \{p\})$  do
24:    if  $\text{SIM}(p, p_i) > \theta$  then
25:       $s \leftarrow s \cup \{p_i\}$ 
26:       $++v_n(s)$ 
27:      if  $r = R_n(p_i)$  then  $++v_n(r, s)$ 
28:       $pr_n(r) \leftarrow \frac{v_n(r, s)}{v_n(s)}$ 
29:       $\text{STORE}(pr_n(r))$ 
30:       $\langle utility(pr_n) \rangle \leftarrow \text{ENCRYPT}(pr_n)$ 
31:      SEND  $\langle utility(pr_n) \rangle$  to  $parent$ 

32: upon receiving  $\langle utility(pr_n) \rangle$  from node  $n$  do
33:  if  $n_i = n_r$  then
34:     $pr_n \leftarrow \text{DECRYPT}(\langle utility(pr_n) \rangle)$ 
35:     $utilities.ADD(pr_n)$ 
36:    if  $expired = \text{true}$  or  $|utilities| = reached$  then
37:      for all  $r \in R$  do
38:         $W(r) \leftarrow \sum_{i=1}^{|utilities|} \log U_i(r)$ 
39:        if  $W(r) > max$  then
40:           $max \leftarrow W(r)$ 
41:           $r_{max} \leftarrow r$ 
42:    else SEND  $\langle utility(pr_n) \rangle$  to  $parent$ 

```

updated utility-values to its parent (Alg. 1 (31)). For privacy reasons, the utility values have been firstly *encrypted* (Alg. 1 (30)) by using the asymmetrical-cryptography technique [14].

Utility message. A node that has received a utility message forwards the message to its parent. All the utility-values are gradually collected to the source node (*convergecast*). The source node decrypts each received utility-message (Alg. 1 (34)) and maintains the utility values (Alg. 1 (35)). Before making a decision, the source node waits until a time limit has expired or all the reachable nodes

Table 1: Statistics for the reported categories of products.

ID	Category	#consumers	#products
A ₁	Beauty	1210271	249274
A ₂	Tools & Home Improvement	1212468	260659
A ₃	Video Games	826767	50210
A ₄	Toys & Games	1342911	327698
A ₅	Health & Personal Care	1851132	252331

have sent utility messages (Alg. 1 (36)). Finally, the source node makes a decision that maximizes the social-welfare function (Alg. 1 (37-41)) – applying Def. 2.1.

3.2 Time and Message Complexity & Scalability

In the broadcast of reach messages, the time complexity is $O(2 * N)$, since all the nodes receive a reach message at most twice. The message complexity is N , since every node sends its own reach message. In the convergecast of source messages, since all the nodes send their source messages to their parents based on a known spinning-tree, the time complexity is $\log N$, where $\log N$ is the depth of the tree. The message complexity is N , since every node sends its own source message. In the broadcast of vote messages, the time and message complexity are N , since all the nodes receive a vote message exactly once. In the convergecast of utility messages, the time complexity is $\log N$ and the message complexity is N . The overall (time and message) complexity of the algorithm, $O(7N + 2\log N) = O(N)$, asymptotically scales in a *linear* way to the total number N of the nodes that are reachable from the source node.

4 PRELIMINARY EVALUATION

Dataset. To evaluate our algorithm on large-scale real-world votes, we used the publicly available⁶ Amazon numerical ratings on thousands of products [15]. We considered that each product corresponds to a voting topic and a rating corresponds to a vote. The products had been organized into high-level categories. Each category contains products that may be similar to each other in terms of their descriptive terms. Due to lack of space, we indicatively present the results for five middle-sized categories⁷ (Table 1).

Evaluation methodology. We implemented a graph simulation of a distributed network and executed our algorithm for each product of the reported categories. At each execution, we configured the network to contain all the consumers who rated an examined product. We also considered a topology of the network such that all the nodes are reachable from at least one source node. To evaluate the fairness of decisions, we present the results for products that have *conflicting* ratings, i.e. there is no rating for a product that represents the majority of consumers. We compared the fairness of the decisions made by our algorithm against the arithmetic mean and median of ratings. These aggregators are the most widely used for making decisions on consumers' choices [16]. We assume that a decision is fair if it represents at least the 60% of the consumers' preferences (i.e. minimum percentage for majority decisions⁸).

⁶ jmcauley.ucsd.edu/data/amazon⁷ We have reached to analogous conclusions for the remaining categories.⁸ <https://www.princeton.edu/~pcwcr/drafting/voting.html>

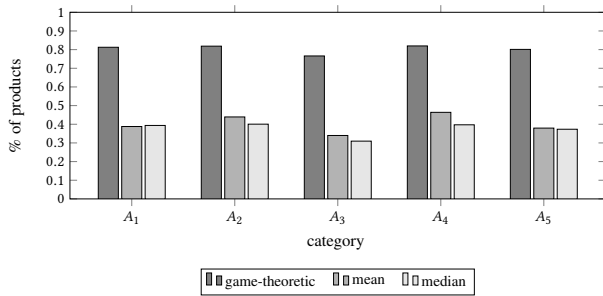


Figure 1: Products with conflicting ratings and fair decisions.

Evaluation results. Fig. 1 depicts for each category the percentages of the products with both conflicting ratings and fair decisions. We observe that the percentages of the fair decisions made by our algorithm range from the 76% to the 81% of products. On the contrary, the corresponding percentages for the arithmetic mean (resp. median) range from the 33% (resp. 30%) to the 46% (resp. 40%) of products. These percentages further show that the decisions made by our algorithm are not the same with arithmetic means and medians in at least the 30% of products (esp. from the 30% to the 41% of products). Concluding, our algorithm makes fair decisions in a high number of cases and is more effective than the typical aggregators.

5 RELATED WORK

From architecture perspective, we organize the existing frameworks for online participatory democracy in three generations: (i) centralized Web-based frameworks (e.g. [17]), (ii) cloud-based frameworks (e.g. Open Town Hall⁹), and service-oriented frameworks (e.g. [2, 18, 19]). Among the five key-functions [2], the existing frameworks have primarily focused on proposal making & voting (e.g. Agora Voting¹⁰), online collaboration & discussion (e.g. Etherpad¹¹), and social communication [19].

Towards supporting the direct democracy, we faced the challenges of the dynamic engagement of citizens and distributed voting. Regarding these challenges, the existing frameworks model citizens by simple user-profiles, without ensuring that relevant citizens are reached (on-demand engagement). Moreover, citizens' data are stored on the server side, raising security and privacy issues. We further faced the challenge of conflicts that emerge in a large number of citizens' votes. Concerning this challenge, the existing frameworks make single-consensus decisions, assuming there are decisions that can represent the majority of votes. Single-consensus decision-making models that have been mainly used are decision trees [20] and Bayesian networks [21].

6 CONCLUSIONS AND FUTURE WORK

To support the direct democracy, we proposed the distributed framework Digital-Ecclesia that offers dynamic citizens' engagement and distributed game-theoretic decisions. We evaluated our framework

on a corpus of real-world choices and the preliminary results showed that our algorithm makes fair decisions in a high number of cases.

A future research-direction is to extend our distributed algorithm with sophisticated security and privacy techniques. On top of this, the voting procedure could be modelled as a cooperative game that gives incentives and rewards to citizens in order to make better choices. A final direction is to extend our framework to support the remaining key-functions of online democracy-frameworks.

REFERENCES

- [1] E. Bruno. Co-deciding with Citizens: Towards Digital Democracy at EU Level. Technical report, European Citizen Action Service (ECAS), 2015.
- [2] J. Holston, V. Issarny, and C. Parra. Engineering software assemblies for participatory democracy: The participatory budgeting use case. In *International Conference on Software Engineering*, pages 573–582, 2016.
- [3] Nicholas C. Kyriazis and Emmanouil Marios L. Economou. *Democracy and Education: A History from Ancient Athens*. Springer International Publishing, 2015.
- [4] K. Arrow. A difficulty in the concept of social welfare. *Journal of Political Economy*, 58, 1950.
- [5] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.
- [6] A. S. Tanenbaum and M. van Steen. *Distributed systems - principles and paradigms, 2nd Edition*. Pearson Education, 2007.
- [7] K. Binmore. *Playing for real: A text on game theory*. 2007.
- [8] P. Coughlin and S. Nitzan. Electoral outcomes with probabilistic voting and nash social welfare maxima. *Journal of Public Economics*, 15(1):113–121, 1981.
- [9] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.
- [10] Rainer Burkard, Mauro Dell'Amico, and Silvano Martello. *Assignment Problems*. Society for Industrial and Applied Mathematics, USA, 2009.
- [11] George A. Miller. Wordnet: a lexical database for english. *ACM Communications*, 38(11):39–41, 1995.
- [12] H. Attiya and J. Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. John Wiley & Sons, 2004.
- [13] Gabriel Valiente. *Algorithms on Trees and Graphs*. Springer, 2002.
- [14] W. Stallings. *Cryptography and Network Security: Principles and Practice*. Pearson Education, 3rd edition, 2002.
- [15] J. McAuley and A. Yang. Addressing complex and subjective product-related queries with customer reviews. In *International Conference on World Wide Web*, pages 625–635, 2016.
- [16] R. Di Pietro, M. Petrocchi, and A. Spognardi. *A Lot of Slots – Outliers Confinement in Review-Based Systems*, pages 15–30. 2014.
- [17] Matt Leighninger. *Using Online Tools to Engage- and be Engaged by- the Public*. Technical report, IBM Center for the Business of Government, 2011.
- [18] A. Pathak, V. Issarny, and J. Holston. Appcivist - A service-oriented software platform for socially sustainable activism. In *International Conference on Software Engineering*, pages 515–518, 2015.
- [19] R. Angarita, N. Georgantas, C. Parra, J. Holston, and V. Issarny. Leveraging the service bus paradigm for computer-mediated social communication interoperability. In *International Conference on Software Engineering*, 2017.
- [20] P. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Pearson Addison Wesley, 2006.
- [21] A. Detwarasiti and R. D. Shachter. Influence diagrams for team decision analysis. *Decision Analysis*, 2(4):207–228, 2005.

⁹<http://opentownhall.com>

¹⁰<http://agoravoting.org>

¹¹<http://etherpad.org>